

```

1  import numpy as np
2  import matplotlib.pyplot as plt
3  import useful as use
4  import pdb
5  from decimal import Decimal
6  np.set_printoptions(threshold=np.inf)
7
8
9  fig1, ax3 = plt.subplots()
10 fig2, ax4 = plt.subplots()
11 fig3, ax5 = plt.subplots()
12 #####
13 #initial conditions
14
15
16 dt = .001
17
18 #####
19 #Theoretical Solution
20
21 l = [1]
22 r_0 = 1
23 v_0 = 0
24 t = 0
25 r = r_0
26 v = v_0
27 s = 2**(1/3)
28 w_0 = 1
29 gamma=.6
30 w = np.sqrt(w_0**2-gamma**2)
31 timing = [0]
32
33 for i in range(0,int(2*2*np.pi/dt)):
34
35     —>r = np.e**(-gamma*t)*(r_0*np.cos(w*t)+(v_0+gamma*r_0)/w*np.sin(w*t))
36     —>
37     —>t+=dt
38     —>l.append(r)
39     —>timing.append(t+dt)
40
41 ax3.plot(timing,l)
42 ax4.plot(timing,l)
43
44 # err_theory = []
45 # count = 0
46 # for i in range(len(l)):
47
48     —># test = count/9000
49     —># if Decimal(test) != Decimal(0):
50     —>—># err_theory.append(l[i])
51     —># count+=1
52 # print(err_theory)
53
54 #####
55 #####
56 #####
57 #First Order
58
59 dt = .9
60 l = [1]
61 r_0 = 1
62 v_0 = 0
63 t = 0
64 w_0 = 1
65 gamma=.6
66 w = np.sqrt(w_0**2-gamma**2)
67 timing = [0]

```

```

68 r = r_0
69 v = v_0
70 s = 2**(1/3)
71
72 for i in range(0,int((2*2*np.pi)/dt)):
73
74     —>r = r+v*dt
75     —>v = v - w_0**2*r*dt
76     —>v = np.e**(-2*gamma*dt)*v
77     —>l.append(r)
78     —>timing.append(timing[len(timing)-1]+dt)
79
80 r = np.asarray(r)
81 ax3.plot(timing,l)
82
83 #####
84 #####
85 #Second Order
86 l = [1]
87 r_0 = 1
88 v_0 = 0
89 t = 0
90 w_0 = 1
91 gamma=.6
92 w = np.sqrt(w_0**2-gamma**2)
93 timing = [0]
94 r = r_0
95 v = v_0
96 s = 2**(1/3)
97
98 for i in range(0,int((4*np.pi)/dt)):
99
100     —>v = v - w_0**2*r*.5*dt
101     —>v = np.e**(-2*gamma*.5*dt)*v
102     —>r = r+v*dt
103     —>v = np.e**(-2*gamma*.5*dt)*v
104     —>v = v - w_0**2*r*.5*dt
105
106     —>l.append(r)
107     —>timing.append(timing[len(timing)-1]+dt)
108
109 ax3.plot(timing,l)
110
111 #####
112 #####
113 #Fourth Order
114 l = [1]
115 r_0 = 1
116 v_0 = 0
117 t = 0
118 w_0 = 1
119 gamma=.6
120 w = np.sqrt(w_0**2-gamma**2)
121 r = r_0
122 v = v_0
123 s = 2**(1/3)
124 H = dt/(2-s)
125 timing = [0]
126
127 for i in range(0,int((4*np.pi)/dt)):
128
129     —>v = v - w_0**2*r*.5*H
130     —>v = np.e**(-2*gamma*.5*H)*v
131     —>r = r+v*H
132     —>v = np.e**(-2*gamma*.5*H)*v
133     —>v = v - w_0**2*r*.5*H
134

```

```

135
136     →v = v - w_0**2*r*.5*-s*H
137     →v = np.e**(-2*gamma*.5*-s*H)*v
138     →r = r+v*-s*H
139     →v = np.e**(-2*gamma*.5*-s*H)*v
140     →v = v - w_0**2*r*.5*-s*H
141
142
143     →v = v - w_0**2*r*.5*H
144     →v = np.e**(-2*gamma*.5*H)*v
145     →r = r+v*H
146     →v = np.e**(-2*gamma*.5*H)*v
147     →v = v - w_0**2*r*.5*H
148
149
150     →l.append(r)
151     →timing.append(timing[len(timing)-1]+dt)
152
153     ax4.plot(timing,l)
154     ax4.set_ylabel('Particle Displacement')
155     ax4.set_xlabel('Time')
156     ax4.legend(('Theory','4th Order'),loc='upper right')
157     ax4.set_title("Damped HO - Comparison with .9 Timestep",va='bottom')
158
159
160     #####
161     #####
162     #graphing
163     ax3.set_ylabel('Particle Displacement')
164     ax3.set_xlabel('Time')
165     ax3.legend(('Theory','1st Order','2nd Order','4th Order'),loc='upper right')
166     # ax3.plot([-0.5,.5],[0,0], 'o', markerfacecolor='none', markeredgecolor='r')
167     ax3.set_title("Damped HO - Comparison with .9 Timestep",va='bottom')
168     plt.show()

```